CLAIMS

We claim:

1. A spoken dialog system having a dialog manager, the dialog manager comprising:

a top level flow controller that manages a process of controlling speech interaction between a spoken dialog system and a user; and

a reusable subdialog that is isolated from application dependencies, wherein the top level flow controller invokes the reusable subdialog to handle a specific task associated with the speech interaction and return control to the top level flow controller after handling the specific task, wherein the reusable subdialog and the top level flow controller operate independent of their decision model.

2. The spoken dialog system of claim 1, wherein the reusable subdialog receives context shift data upon being called up by the top level flow controller and returns data to the top level flow controller associated with its interaction with the user.

3. The spoken dialog system of 2, further comprising a plurality of top level flow controllers, each of the plurality of top level flow controllers having at least one reusable subdialog associated with it.

4. The spoken dialog system of 3, wherein the decision model of the top level flow controller is a recursive transition network.

5. The spoken dialog system of 3, wherein the decision model of the top level flow controller is a rule-based model.

6. The spoken dialog system of 1, wherein the top level flow controller supports context shifts.

7. The spoken dialog system of 6, wherein context shifts are triggered by input information from a user and generate a name of a destination state.

8. The spoken dialog system of 1, wherein when a top level flow controller invokes a subdialog, the subdialog inherits the context shifts of the top level flow controller.

9. A method of switching contexts within a spoken dialog between a user and a spoken dialog system, the spoken dialog system having a dialog manager with a first flow controller and a second flow controller, the method comprising:

while the spoken dialog is being controlled by the first flow controller, receiving context-changing input associated with speech from a user that changes a dialog context;

comparing the context-changing input to at least one context shift;

if any of the context shifts are activated by the comparing step, then passing control to an invoked second flow controller indicated by the context shift; and

if no context shift is activated by the comparing step, then maintaining control of the spoken dialog with the first flow controller.

10. The method of claim 9, wherein the at least one context shift further comprises a table of context shifts.

11. The method of claim 10, further comprising storing a local context associated with each of the first and second flow controllers, the local context maintaining a state of the flow controller that is independent of implemented subdialogs.

12. The method of claim 11, wherein the second flow controller receives data values stored in the local context of the first flow controller.

13. The method of claim 9, further comprising maintaining a stack of flow controllers, wherein each invoked flow controller is added to the stack of flow controllers.

14. The method of claim 13, wherein each invoked flow controller inherits a context shift and becomes the recipient of all user input as part of the spoken dialog interaction until the invoked flow controller relinquishes control of the spoken dialog.

15. The method of claim 10, further comprising storing a local context associated with each of the first and second flow controllers, the local context maintaining a state of the flow controller that is independent of implemented subdialogs.

16. A computer-readable medium for storing computer instructions for controlling a computing device to switch contexts within a spoken dialog between a user and a spoken dialog system, the spoken dialog system having a dialog manager with a first flow controller and a second flow controller, the computer instructions comprising the steps:

while the spoken dialog is being controlled by the first flow controller, receiving context-changing input associated with speech from a user that changes a dialog context;

comparing the context-changing input to at least one context shift;

if any of the context shifts are activated by the comparing step, then passing control to an invoked second flow controller indicated by the context shift; and

if no context shift is activated by the comparing step, then maintaining control of the spoken dialog with the first flow controller.

17. The computer-readable medium of claim 16, wherein the at least one context shift further comprises a table of context shifts.

18. The computer-readable medium of claim 17, wherein the steps further comprise storing a local context associated with each of the first and second flow controllers, the

local context maintaining a state of the flow controller that is independent of implemented subdialogs.

19. The computer-readable medium of claim 18, wherein the second flow controller receives data values stored in the local context of the first flow controller.

20. The computer-readable medium of claim 16, wherein the steps further comprise maintaining a stack of flow controllers, wherein each invoked flow controller is added to the stack of flow controllers.

21. The computer-readable medium of claim 20, wherein each invoked flow controller inherits context shifts and becomes the recipient of all user input as part of the spoken dialog interaction until the invoked flow controller relinquishes control of the spoken dialog.

22. The computer-readable medium of claim 17, wherein the steps further comprise storing a local context associated with each of the first and second flow controllers, the local context maintaining a state of the flow controller that is independent of implemented subdialogs.

23. A dialog manager for use in a spoken dialog system, the dialog manager comprising:

means for managing a process of controlling speech interaction between a spoken dialog system and a user; and

a reusable subdialog that is isolated from application dependencies, wherein the means for managing the process of controlling speech interaction invokes the reusable subdialog to handle a specific task associated with the speech interaction and return control to the means for managing the process of controlling speech interaction after handling the specific task.

24. The dialog manager of claim 23, wherein the reusable subdialog receives context data upon being called up by the means for managing the process of controlling speech interaction and returns data associated with its interaction with the user.

25. The dialog manager of 24, further comprising a plurality of means for managing the process of controlling speech interaction, each of the means for managing the process of controlling speech interaction having at least one reusable subdialog associated with it.

26. The dialog manager of 25, wherein the means for managing the process of controlling speech interaction is a recursive transition network.

27. The dialog manager of 25, wherein the means for managing the process of controlling speech interaction is a rule-based model.

28. The dialog manager of 25, wherein the means for managing the process of controlling speech interaction supports context shifts.

29. The dialog manager of 28, wherein context shifts are triggered by input information and a name of a destination state.

30. The dialog manager of 28, wherein when a means for managing the process of controlling speech interaction calls up a subdialog, the subdialog inherits the context shifts of the means for managing the process of controlling speech interaction.

31. A method of switching contexts within a spoken dialog between a user and a spoken dialog system, the spoken dialog system having a dialog manager that manages a stack of dialog modules during a spoken dialog with a user, the method comprising:

when a parent dialog module on the top of the stack of dialog modules that is managing the spoken dialog with a user invokes a subdialog module to manage a portion of a spoken dialog with a user, pushing the subdialog onto the top of the stack, wherein the subdialog manages the spoken dialog with the user and wherein the subdialog is isolated from application dependencies;

transmitting context shift transition data to the pushed subdialog;

if the user changes the context of the dialog while the dialog is being managed by the subdialog:

returning a context shift message and a destination state to the parent dialog;

popping the subdialog off the stack of dialog modules; and

returning management of the spoken dialog to the parent dialog.

32. The method of claim 31, wherein the state of the dialog at any moment is determined according to a decision model of the dialog module currently managing the spoken dialog.

33. The method of claim 31, wherein information is transmitted and received from an invoked subdialog via local memory.

34. The method of claim 31, wherein information is transmitted and received from an invoked subdialog via global memory.

35. The method of claim 31, wherein if the user changes the context of the dialog while the dialog is being managed by the subdialog, the method further comprises popping all subdialogs off the stack and returning control of the spoken dialog to the parent dialog module.